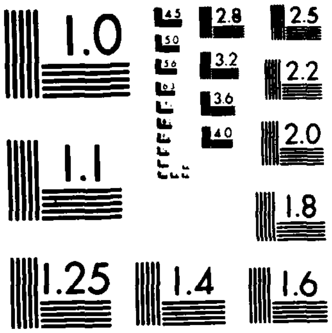
AD-A193 470 AN AUTOMATIC PROGRAM SYNTHESIS BIBLIOGRAPHY(U) NEVADA 1/1
UNIV LAS VEGAS DEPT OF COMPUTER SCIENCE AND ELECTRICAL
.. T A GROSS ET AL. 19 JAN 88 CSR-88-01
UNCLASSIFIED ARO-24960.7-MA-REP DAAL03-87-G-0004 F/G 12/5 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

An Automatic Program Synthesis Bibliography

Todd A Gross
Thomas A Nartker*
Department of Computer Science and Electrical Engineering
University of Nevada, Las Vegas

January 19, 1988

Department of Computer Science and Electrical Engineering

University of Nevada, Las Vegas
Las Vegas, Nevada 89154

2

An Automatic Program Synthesis Bibliography

Todd A Gross
Thomas A Nartker*
Department of Computer Science and Electrical Engineering
University of Nevada, Las Vegas

January 19, 1988

Report CSR-88-01

*Supported by the U. S. Army Research Office under Grant DAAL03-87-G-0004

UNCLASSIFIED MASTER COPY - FOR REPRODUCTION PURPOSES

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
ARO 24960.7-MA-REP

6a. NAME OF PERFORMING ORGANIZATION
Univ. of Nevada

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
U. S. Army Research Office

6c. ADDRESS *(City, State, and ZIP Code)*
Las Vegas, Nevada 89154

7b. ADDRESS *(City, State, and ZIP Code)*
P. O. Box 12211
Research Triangle Park, NC 27709-2211

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
U. S. Army Research Office

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
DAAL03-87-G-0004

8c. ADDRESS *(City, State, and ZIP Code)*
P. O. Box 12211
Research Triangle Park, NC 27709-2211

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
An Automatic Program Synthesis Bibliography

12. PERSONAL AUTHOR(S) Todd A. Gross and Thomas A. Nartker

13a. TYPE OF REPORT
Technical

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Year, Month, Day)*
Jan 19, 1988

15. PAGE COUNT
22

16. SUPPLEMENTARY NOTATION The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Bibliography, Program Synthesis, Automatic Program Synthesis

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The bibliography contains 262 references.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED/UNLIMITED □ SAME AS RPT. □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE *(Include Area Code)*

22c. OFFICE SYMBOL

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

References

[1] A Adam. *GADGET—un Programme de Generation Automatique de Programs sur les Graphes et les Ensembles*. PhD thesis, University of Paris VI, 1975.

[2] S Amarel. *Self Organizing Systems*, pages 107–175. Spartan, New York, 1962. titled On the Automatic Formation of a Computer Program which Represents a Theory.

[3] D Angluin. Summers revisited. 1981. Internal Report, Yale University.

[4] D Angluin and C H Smith. A discrete theory of automatic program synthesis. 1981. In preparation.

[5] G Degli Antoni, P A Miglioli, and M Ornaghi. Top-down approach to the synthesis of programs. In *Proceedings of the Programming Symposium*, pages 88–108, Apr 1974.

[6] Automatic Programming Group. *Project MAC Progress Report X*. Technical Report, Massachusetts Institute of Technology, Cambridge, MA, Jul 1973. Page 172-191.

[7] Automatic Programming Group. *Project MAC Progress Report XI*. Technical Report, Massachusetts Institute of Technology, Cambridge, MA, Jul 1974. Page 107-??

[8] R Balzer. A 15 year perspective on automatic programming. *IEEE Transactions on Software Engineering*, SE-11(11):1257–1268, 1985.

[9] R Balzer. *Automatic Programming*. Technical Report RR-73-1, USC Information Sciences Institute, Marina Del Rey, CA, Sep 1972.

[10] R Balzer. A global view of automatic programming. In *Proceedings of the Third International Joint Conference on Artificial Intelligence*, pages 494–499, Aug 1973.

[11] R Balzer. Transformational implementation: an example. *IEEE Transactions or. Software Engineering*, SE-7(1):3–14, 1981.

[12] R Balzer, N Goldman, and D Wile. On the transformational implementation approach to programming. In *Proceedings of the Second International Conference on Software Engineering*, pages 337–344, Oct 1976.

[13] R Balzer, N Greenfeld, M Kay, W Mann, W Ryder, and A Zobrist. Domain-independent automatic programming. In J L Rosenfeld, editor, *Proceedings of IFIP Congress 1974*, pages 326–330, American Elsevier, New York, 1974.

[14] D Barstow. A knowledge-based system for automatic program construction. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, Aug 1977.

[15] D Barstow. The roles of knowledge and deduction in program synthesis. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 37–43, Aug 1979.

[16] D Barstow, R Duffey, S Smoliar, and S Vestal. An automatic programming system to support an experimental science. In *Proceedings of the Sixth International Conference on Software Engineering*, pages 360–366, Sep 1982.

[17] D Barstow and E Kant. Observations on the interactions between coding and efficiency knowledge in the PSI system. In *Proceedings of the Second International Conference on Software Engineering*, pages 19–31, 1977.

[18] D R Barstow. *Automatic Construction of Algorithms and Data Structures Using A Knowledge Base of Programming Rules.* PhD thesis, Stanford University, Stanford, CA, 1977. Stanford AI Memo AIM-308.

[19] D R Barstow. Automatic programming for streams. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Aug 1985.

[20] D R Barstow. Domain-specific automatic programming. *IEEE Transactions on Software Engineering*, SE-11(11):1321–1336, 1985.

[21] D R Barstow. *Knowledge-Based Program Construction.* North-Holland, New York, 1979.

[22] D R Barstow. A perspective on automatic programming. *Artificial Intelligence Magazine*, 5(1):5–27, 1984.

[23] D R Barstow. The role of knowledge and deduction in algorithm creation. In *International Workshop on Program Construction*, Sep 1980.

[24] J Barzdin. *On Synthesizing Programs Given by Examples.* Technical Report, Computing Center, Latvian State University, Riga, Latvia USSR, 1972.

[25] J M Barzdin. Inductive inference of automata, functions, and programs. *American Mathematical Society Translations*, 109(2):107–112, 1977.

[26] F L Bauer. From specifications to machine code: program construction through formal reasoning. In *Proceedings of the Sixth International Conference on Software Engineering*, Sep 1982.

[27] F L Bauer, M Broy, R Gnatz, W Hesse, and B Krieg-Bruckner. *Notes on the Project CIP: Outline of a Transformation System.* Technical Report TUM-INFO-7729, Technische Universitat Munchen, Munich, Jul 1977.

[28] M A Bauer. Programming by examples. *Artificial Intellligence*, 12(1):1–21, 1979.

[29] W Bibel. Logical program synthesis. In T Moto-oka, editor, *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 227–236, North-Holland, Amsterdam, 1982. Held in Tokyo in 1981.

[30] W Bibel. Syntax-directed semantics-supported program synthesis. *Artificial Intelligence*, 14:243–261, 1980.

[31] W Bibel, U Furbach, and J F Schreiber. *Programmiersprachen*, page . Volume 12 of *Informatik-Fachberichte*, Springer-Verlag, Berlin, 1978.

[32] W Bibel and K M Hornig. *Automatic Program Construction Techniques*, chapter 3. Macmillan, New York, 1984. titled LOPS, A System Based on a Strategical Approach to Program Synthesis.

[33] M Bidoit, B Biebow, M-C Gaudel, C Gresse, and G Guiho. Exception handling: formal specification and systematic program construction. *IEEE Transactions on Software Engineering*, SE-11(3):242–251, 1985.

[34] M Bidoit, C Gresse, and G Guiho. A system which synthesises array manipulating programs from specifications. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, Aug 1979.

[35] A Biermann, G Guiho, and Y Kodratoff. *Automatic Program Construction Techniques*, chapter 1. Macmillan, New York, 1984. titled An Overview of Automatic Program Construction Techniques.

[36] A W Biermann. *Advances in Computers*, pages 1–63. Volume 15, Academic Press, New York, 1976.

[37] A W Biermann. *Automatic Program Construction Techniques*, chapter 17. Macmillan, New York, 1984. titled Dealing With Search.

[38] A W Biermann. Automatic programming: a tutorial on formal methodologies. *Journal of Symbolic Computation*, 1:119–142, 1986.

[39] A W Biermann. Computer program synthesis from computation traces. In *Proceedings of the Symposium on Fundamental Theory of Programming*, pages 89–100, Oct 1972.

[40] A W Biermann. *Proceedings of European Conference on Computer Algebra (EUROCAL 85)*, pages 124–135. Volume 203 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1985. Conference in Linz, Austria in Apr 85.

[41] A W Biermann. Regular LISP programs and their automatic synthesis from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-8(8):585–600, 1978.

[42] A W Biermann, R I Baum, R Krishnaswamy, and F Petry. Autoprogrammer. Feb 1974. 10-minute silent movie.

[43] A W Biermann, R I Baum, and F Petry. Speeding up the synthesis of programs from traces. *IEEE Transactions on Computers*, C-24(2):122–136, 1975.

[44] A W Biermann and R Krishnaswamy. Constructing programs from example computations. *IEEE Transactions on Software Engineering*, SE-2(3):141–153, 1976.

[45] A W Biermann and R Krishnaswamy. A system for program synthesis from examples. In *Proceedings of IEEE Systems, Man, and Cybernetics Conference*, Oct 1974.

[46] A W Biermann and D R Smith. *The Hierarchical Synthesis of LISP Scanning Programs*, pages 41–45. North-Holland, 1977.

[47] A W Biermann and D R Smith. A production rule mechanism for generating LISP code. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-9(5):260–276, 1979.

[48] T J Biggerstaff. *Automatic Program Construction Techniques*, chapter 18. Macmillan, New York, 1984. titled Design Directed Synthesis of LISP Programs.

[49] T J Biggerstaff. *C2: A 'Super Compiler' Approach to Automatic Programming.* PhD thesis, University of Washington, Seattle, WA, Jan 1976.

[50] T J Biggerstaff and D L Johnson. *Design Directed Program Synthesis.* Technical Report 77-02-01, Department of Computer Science, University of Washington, Seattle, WA, Feb 1977.

[51] B I Blum. Experience with an automated generator of information systems. In *1985 IEEE International Symposium on New Directions in Computing*, pages 138–147, 1985.

[52] B I Blum. An information system for developing information systems. In *1983 National Computer Conference*, pages 743–752, 1983.

[53] B I Blum. A MUMPS program generator with a relational database manager. In , MUMPS Users' Group, Jun 1981.

[54] B I Blum. A tool for developing information systems. In A Wasserman, editor, *Proceedings of the IFIP WG8.1 Working Conference on Automated Tools for Information Systems Design and Development*, pages 215–235, North-Holland, Jan 1982.

[55] M A Breuer. Generation of optimal code for expressions via factorization. *Communications of the ACM*, 12(6):333–340, 1969.

[56] R H Brown. Automatic synthesis of numerical computer programs. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 998–1003, Aug 1981.

[57] R H Brown. Automation of programming; the ISFI experiments. In *1985 IEEE Symposium on Expert Systems in Government Systems*, pages 232–238, 1985.

[58] R H Brown. *Coherent Behavior from Incoherent Knowledge Sources in the Automatic Synthesis of Numerical Computer Programs.* Technical Report MIT-AI-TR-610, Massachusetts Institute of Technology, Cambridge, MA, Jan 1981.

[59] M Broy. Program construction by transformations: a family tree of sorting programs. In A W Biermann and G Guiho, editors, *Proceedings of the NATO Advanced Study Institute*, pages 1–49, D Reidel, Sep-Oct 1981.

[60] J R Buchanan. *A Study in Automatic Programming.* Technical Report, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA, 1974.

[61] J R Buchanan and D C Luckham. *On Automating the Construction of Programs.* Technical Report, Stanford Artificial Intelligence Laboratory, Stanford, CA, May 1974.

[62] R Burstall and M Feather. Program development by transformations: an overview. In *Proceedings of Toulouse CREST Course on Programming*, 1978.

[63] R M Burstall and J Darlington. Some transformations for developing recursive programs. In *Proceedings of 1975 International Conference on Reliable Software*, pages 465–472, 1975.

[64] R M Burstall and J Darlington. A transformation system for developing recursive programs. *Journal of the Association for Computing Machinery*, 24(1):44–67, 1977.

[65] J Cadiou and Z Manna. Recursive definitions of partial functions and their computations. *SIGPLAN Notices*, 7:58–65, 1972.

[66] F Caio, G Guida, and M Somalvico. Problem solving as a basis for program synthesis: design and experimentation of the BIS system. *International Journal of Man-Machine Studies*, 17:173–188, 1982.

[67] A F Cardenas. Technology for automatic generation of application programs—a pragmatic view. *MIS Quarterly*, Sep 1977.

[68] T E Cheatham Jr and B Wegbreit. A laboratory for the study of automatic programming. In *Proceedings of the AFIPS 1972 SJCC*, pages 11–21, AFIPS Press, 1972. volume 40.

[69] T T Cheng, E D Lock, and N S Prywes. Use of very high level languages and program generation by management professionals. *IEEE Transactions on Software Engineering*, SE-10(5):552–563, 1984.

[70] K L Clark. *The Synthesis and Verification of Logic Programs.* Research Report DOC 81/36, Department of Computing, Imperial College, London, UK, 1981.

[71] K L Clark and J Darlington. Algorithm classification through synthesis. *Computer Journal*, 23(1):61–65, 1980.

[72] K L Clark and S Sickel. Predicate logic: a calculus for deriving programs. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 419–420, Aug 1977.

[73] B Cohen and C Sammut. *Automatic Program Construction Techniques*, chapter 21. Macmillan, New York, 1984. titled Program Synthesis Through Concept Learning.

[74] N H Cohen. Mechanical analysis and restructuring of recursive programs. May 1976. Internal Report, Harvard University.

[75] R L Constable. *Information Processing 71*, pages 733–738. North-Holland, Amsterdam, 1972. titled Constructive Mathematics and Automatic Program Writers.

[76] Y Damghani and T G Cleaver. Courseware authoring tools. In *Conference Proceedings of IEEE SOUTHEASTCON 84*, pages 143–147, IEEE, New York, Apr 1984.

[77] J Darlington. Applications of program transformation to program synthesis. In *Colloques IRIA sur la Construction, L'Amelioration et la Verification Systematique des Programmmes*, pages 133–144, 1975.

[78] J Darlington. *Program Transformation and Synthesis. Present Capabilities.* Technical Report 77/43, Department of Computing and Control, Imperial College, London, UK, 1977.

[79] J Darlington. The synthesis of implementations for abstract data types. In A W Biermann and G Guiho, editors, *Proceedings of the NATO Advanced Study Institute*, pages 309–334, D Reidel, Sep-Oct 1981.

[80] J Darlington. A synthesis of several sorting algorithms. *Acta Informatica*, 11(1):1–30, 1978.

[81] J L Darlington. Automatic program synthesis in second-order logic. In *Advance Papers of the Third International Joint Conference on Artificial Intelligence*, pages 479–485, Aug 1973.

[82] U De Carlini, A Esposito, and B Fadini. Automatic generation of application programs on minicomputers. *International Journal of Mini and Microcomputers*, 6(2):21–25, 1984.

[83] N M Delisle, D E Menicosy, and N L Kerth. Tools for supporting structured analysis. In A Wasserman, editor, *Proceedings of the IFIP WG8.1 Working Conference on Automated Tools for Information Systems Design and Development*, pages 11–20, North-Holland, 1982.

[84] N Dershowitz. Synthetic programming. *Artificial Intelligence*, 25(3):323–373, 1985.

[85] N Dershowitz and Z Manna. On automating structured programming. In *Proceedings Colloques IRIA sur la Construction, L'Amelioration et la Verification Systematique des Programmes*, Jul 1975.

[86] P Deschamp. *Production de Compilateurs a Partir d'une Description Simantique des Langages de Programmation: le Systeme Perluette.* PhD thesis, INPL, Mar 1980. These de Docteur-Ingenieur.

[87] P Dewan. *Automatic Generation of User Interfaces.* PhD thesis, University of Wisconsin, Madison, WI, Aug 1986.

[88] P Dewan and M Solomon. Dost: an environment to support automatic generation of user interfaces. In *Proceedings of the ACM SIGSOFT/SIGPLAN Software Symposium on Practical Software Development Environments*, pages 150–159, Dec 1986.

[89] J W Duran. *A Study of Loop Invariants and Automatic Program Synthesis.* Technical Report SESLTR-12, Department of Computer Science, University of Texas, Austin, TX, 1975.

[90] J Earley. High level operations in automatic programming. In *Proceedings of Symposium on Very High Level Languages*, pages 34–42, 1974.

[91] S L Ehrenreich and W A Harris. JMOS: stepping outside with new cost control. *Bell Laboratories Record*, Jul 1985.

[92] R Elsclager and J Phillips. *Automatic Programming.* Technical Report STAN-CS-79-578, Stanford Artificial Intelligence Laboratory, Stanford, CA, Aug 1979.

[93] R Balzer et al. *Domain-Independent Automatic Programming*, page 326. Volume 74 of *Information Processing*, North-Holland, Amsterdam, 1974.

[94] M Feather. *A System for Developing Programs by Transformations.* PhD thesis, University of Edinbergh, 1979.

[95] M S Feather. *ZAP Program Transformation System, Primer and User's Manual.* Technical Report #54, Department of Artificial Intelligence, University of Edinbergh, 1979.

[96] J A Feldman. *Automatic Programming.* Technical Report 255, Department of Computer Science, Stanford University, Stanford, CA, Feb 1972.

[97] J A Feldman. Some decidability results on grammatical inference of best programs. *Information and Control*, 20:244–262, 1972.

[98] S Fickas. Automatic goal-directed program transformation. In *Proceedings of the First National Conference on Artificial Intelligence*, 1980.

[99] S Fickas. *Automating the Transformational Development of Software.* PhD thesis, University of California, Irvine, CA, 1982.

[100] S F Fickas. Automating the transformational development of software. *IEEE Transactions on Software Engineering*, SE-11(11):1268–1277, 1985.

[101] W Finzer and L Gould. *Programming by Rehearsal.* Memo SCL-84-1, Palo Alto Research Center, May 1984. Sort version in BYTE v9 #6 June 1984.

[102] R W Floyd. Towards interactive design of correct programs. In *Proceedings of IFIP Congress*, pages 7 10, North-Holland, 1967. Volume 1.

[103] R Follett. *Automatic Program Construction Techniques*, chapter 4. Macmillan, New York, 1984. titled Combining Program Synthesis With Program Analysis.

[104] R Follett. *Automatic Program Synthesis.* PhD thesis, University of New South Wales, 1980.

[105] R Follett. Combining program synthesis with program analysis. In *Proceedings of the International Workshop on Program Construction*, Sep 1980.

[106] R Follett. Describing the complete effects of programs. In *Proceedings of the Symposium on Language Design and Programming Methodology*, pages 95–104, Sep 1979.

[107] R Follett. *PROSYN—A Program Synthesiser.* Internal Report, University of New South Wales, 1979.

[108] R Follett. Synthesizing recursive functions with side effects. *Artificial Intelligence*, 13(3):175–200, 1980.

[109] J L Gana. *Use and Extension of an Automatic Program Generator for Model Building in Social and Engineering Sciences.* PhD thesis, University of Pennsylvania, Philadelphia, PA, May 1978.

[110] B L Gates and J A van Hulzen. *Proceedings of European Conference on Computer Algebra (EUROCAL 85)*, pages 583–584. Volume 203 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1985. Conference in Linz, Austria in Apr 85.

[111] B L Gates and P S Wang. A LISP-based RATFOR code generator. In *Proceedings of the 1984 MACSYMA Users Conference*, pages 319–329, 1984.

[112] R Gerritsen. *Automatically Generated Programs for Information Retrieval.* Technical Report, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, 1973.

[113] G Gini. The automatic synthesis of iterative programs. *Information Processing Letters*, 14(2):67–73, 1982.

[114] J Ginsparg. *Natural Language Processing in an Automatic Programming Domain.* Technical Report AIM-316, Stanford University, Jun 1978.

[115] P C Goldberg. *Programming Methodology*, chapter . Volume 23 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1975. titled Automatic Programming.

[116] S Goto. Program synthesis from natural deduction proofs. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 339–341, Aug 1979.

[117] S Goto. Program synthesis through godel's interpretation. In *Proceedings of the International Conference on Mathematical Studies of Information Processing*, pages 287–306, 1978.

[118] C Green, R Gabriel, E Kant, B Kedziersk, B McCune, J Phillips, S Tappel, and S Westfold. Results in knowledge-based program synthesis. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 342–344, Aug 1979.

[119] C Green, J Phillips, S Westfold, T Pressburger, B Kedzierski, S Angebranndt, B Mont-Reynaud, and S Tappel. *Research on Knowledge-Based Programming and Algorithm Design—1981.* Technical Report KES-U-81-2, Kestrel Institute, Palo Alto, CA, 1981.

[120] C Green and S Westfold. Knowledge-based programming self applied. *Machine Intelligence*, 10, 1982.

[121] C C Green. The design of the PSI program synthesis system. In *Proceedings of the Second International Conference on Software Engineering*, pages 4–18, Oct 1976.

[122] C C Green. A summary of the PSI program synthesis system. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 380–381, Aug 1977.

[123] C C Green and D R Barstow. On program synthesis knowledge. *Artificial Intelligence*, 10:241–279, Nov 1978.

[124] C C Green and D R Barstow. Some rules for the automatic synthesis of programs. In *Advanced Papers of the Fourth International Joint Conference on Artificial Intelligence*, pages 232–239, Sep 1975.

[125] C Gresse. Automatic programming for data types decomposition patterns. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, 1983.

[126] G Guiho and C Gresse. Program synthesis from incomplete specifications. In *Proceedings of the Fifth Conference on Automated Deduction*, pages 53–62, 1980.

[127] M Hammer and G Ruth. *Research Directions in Software Technology*, chapter 20. MIT Press, Cambridge, MA, 1979. titled Automating the Software Development Process.

[128] J Hardy. Synthesis of LISP functions from examples. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, pages 268–273, Sep 1975.

[129] A C Hax and W A Martin. Automatic generation of customized, model-based information systems for operations management. *Data Base*, 140–144, Winter 1973.

[130] G Heidorn. *Automatic Programming Through Natural Language Dialogue.* Technical Report, IBM Thomas J Watson Research Center, Yorktown Heights, NY, 1975.

[131] G Heidorn. *Natural Language Inputs to a Simulation Programming System.* Technical Report, Naval Postgraduate School, Monterey, CA, 1972.

[132] G Heidorn. *Supporting a Computer Directed Natural Language Dialogue for Automatic Business Programming.* Technical Report RC-6041 (#26157), IBM Thomas J Watson Research Center, Yorktown Heights, NY, 1976.

[133] G E Heidorn. Automatic programming through natural language dialogue: a survey. *IBM Journal of Research and Development*, 20(4):302–313, 1976.

[134] C J Hogger. Derivation of logic programs. *Journal of the Association for Computing Machinery*, 28(2):372–392, 1981.

[135] C J Hogger. *Derivation of Logic Programs.* PhD thesis, University of London, 1979.

[136] C J Hogger. Goal-directed derivation of logic programs. In *Proceedings of the Seventh Symposium on Mathematical Foundations of Computer Science*, 1978.

[137] C J Hogger. Program synthesis in predicate logic. In *Proceedings of the AISB/GI Conference on Artificial Intelligence*, 1978.

[138] S Horwitz. *Generating Language-Based Editors: A Relationally-Attributed Approach.* PhD thesis, Cornell University, Ithaca, NY, Aug 1985.

[139] S Horwitz and T Teitelbaum. Generating editing environments based on relations and attributes. *ACM Transactions on Programming Languages and Systems*, 8(4):577–608, 1986.

[140] G W Howe, V J Kruskal, and I Wladawsky. *A New Approach for Customizing Business Applications.* Technical Report RC5474 (#23258), IBM Thomas J Watson Research Center, Yorktown Heights, NY, 1975.

[141] J Hsiang. *Topics in Automated Theorem Proving and Program Construction.* PhD thesis, University of Illinois, Urbana, IL, 1982.

[142] G Huet and B Lang. Proving and applying program transformations expressed with second-order patterns. *Acta Informatica*, 11(1):31–55, 1978.

[143] J-P Jouannaud and G Guiho. Inference of functions with an interactive system. *Machine Intelligence*, 9, 1977.

[144] J-P Jouannaud and G Guiho. *Program Synthesis from Examples for a Simple Class of Non-Loop Functions.* Technical Report, Laboratoire de Recherche en Informatique, Universite de Paris-Sud, 1977.

[145] J-P Jouannaud, G Guiho, and J-P Treuil. SISP/1, an interactive system able to synthesize functions from examples. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 412–418, Aug 1977.

[146] J-P Jouannaud and Y Kodratoff. An automatic construction of LISP programs by transformations of functions synthesized from their input-output behavior. *International Journal of Policy Analysis and Information Systems*, 4:331–358, 1980.

[147] J-P Jouannaud and Y Kodratoff. Characterization of a class of functions synthesized from examples by a summers-like method using a boyer-moore-wegbreit matching technique. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 440–447, Aug 1979.

[148] J-P Jouannaud and Y Kodratoff. Program synthesis from examples of behavior. In A W Biermann and G Guiho, editors, *Proceedings of the NATO Advanced Study Institute*, pages 213–250, D Reidel, Sep-Oct 1981.

[149] E Kant. *Efficiency Considerations in Program Synthesis: A Knowledge-Based Approach.* PhD thesis, Stanford University, Stanford, CA, Sep 1979.

[150] E Kant. *Efficiency in Program Synthesis.* UMI Research Press, Ann Arbor, MI, 1981.

[151] E Kant. On the efficient synthesis of efficient programs. *Artificial Intelligence*, 20(3):253–305, 1983.

[152] E Kant. The selection of efficient implementations for a high-level language. In *Proceedings of the ACM SIGART-SIGPLAN Symposium on Artificial Intelligence and Programming Languages*, Aug 1977.

[153] E Kant. Understanding and automating algorithm design. *IEEE Transactions on Software Engineering*, SE-11(11):1361–1374, 1985.

[154] E Kant and D R Barstow. *Interactive Programming Environments*, pages 487–513. McGraw-Hill, New York, 1984. titled The Refinement Paradigm: The Interaction of Coding and Efficiency Knowledge in Program Synthesis.

[155] E Kant and A Newell. An automatic algorithm designer: an initial implementation. In *Proceedings of the American Association of Artificial Intelligence*, 1983.

[156] U Kastens, B Hutt, and E Zimmerman. *GAG: A Practical Compiler Generator*. Volume 141 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1982.

[157] A I Kazmin, A A Menn, and N N Nepeivoda. Table approach tc automatic program synthesis. *Programmirovanie*, (2):24-34, Mar-Apr 1982. In Russian, translated by Plenum Publishing in 1983.

[158] Y Kodratoff. Choix d'un programme LISP correspondant a un exemple. In *Congres AFCET-IRIA Reconnaissance des Formes et Traitement des Images*, pages 212–219, 1978.

[159] Y Kodratoff. A class of functions synthesized from a finite number of examples and a LISP program scheme. *International Journal of Computer and Information Sciences*, 8:489–521, 1979.

[160] Y Kodratoff and J Fargues. A sane algorithm for the synthesis of LISP functions from example problems: the boyer-moore algorithm. In *Proceedings of the AISB Meeting*, pages 169–175, 1978.

[161] Y Kodratoff and J-P Jouannaud. *Automatic Program Construction Techniques*, chapter 16. Macmillan, New York, 1984. titled Synthesizing LISP Programs Working on the List Level of Embedding.

[162] Y Kodratoff and E Papon. A system for program synthesis and program optimization. In *Proceedings of AISB*, pages 1–10, 1980.

[163] G Kreisel. Some uses of proof theory for finding computer programs. In *Colloque International de Logique*, pages 123–134, 1977.

[164] M Kyllastinen. Aspects about automatic programming. In *Proceedings of NORDATA 81*, Dansk Databehandlingsforening, Copenhagen, Denmark, Jun 1981.

[165] W T Laaser. *Synthesis of Recursive Programs*. PhD thesis, Stanford University, Stanford, CA, 1979.

[166] S S Lavrov. Program synthesis. *Kibernetika*, (6):11-16, Nov-Dec 1982. in Russian, translated in 1983 by Plenum Publishing.

[167] R C Lee, C L Chang, and R J Waldinger. An improved program-synthesizing algorithm and its correctness. *Communications of the ACM*, 17(4):211–217, 1974.

[168] R C Lee and S K Chang. Structured programming and automatic program synthesis. In *Proceedings of the Symposium on Very High Level Languages*, pages 60–70, Mar 1974.

[169] E Levinson, L S Levy, and J B Salisbury. CARL—experience of an application using clusters. In *Proceedings of NCC*, 1980.

[170] L S Levy. A metaprogramming method and its economic justification. *IEEE Transactions on Software Engineering*, SE-12(2):272–277, 1986.

[171] W J Long. *A Program Writer*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1977. Report LCS/TR-187.

[172] J Low. *Automatic Coding: Choice of Data Structures*. Technical Report AIM-242, Stanford University, Stanford, CA, Aug 1974.

[173] C J Lucena and T H Pequeno. Program derivation using data types: a case study. *IEEE Transactions on Software Engineering*, SE-5(6):586–592, 1979.

[174] D C Luckham and J R Buchanan. Automatic generation of programs containing conditional statements. In *Proceedings of Conference on Artificial Intelligence and the Simulation of Behavior*, pages 102–126, Jul 1974.

[175] P A Luker. MODELLER—computer-assisted modelling of continuous systems. *Simulation*, 42(5), 1984.

[176] P A Luker and A Burns. Program generators and generation software. *Computer Journal*, 29(4):315–321, 1986.

[177] P A Luker and J Stephenson. *Simulation of Systems*, page . North-Holland, Amsterdam, 1979.

[178] Z Manna and R Waldinger. The automatic synthesis of recursive programs. In *Proceedings of the Symposium on Artificial Intelligence and Programming Languages*, 1977.

[179] Z Manna and R Waldinger. A deductive approach to program synthesis. *ACM Transactions on Programming Languages and Systems*, 2(1):90–121, 1980.

[180] Z Manna and R Waldinger. Knowledge and reasoning in program synthesis. *Artificial Intelligence Journal*, 6(2):175–208, 1980.

[181] Z Manna and R Waldinger. Synthesis: dreams ⇒ programs. *IEEE Transactions on Software Engineering*, SE-5(4):294–328, 1979.

[182] Z Manna and R Waldinger. Toward automatic program synthesis. *Communications of the ACM*, 14(3):151-165, 1971.

[183] W A Martin. OWL, a system for building expert problem solving systems involving verbal reasoning. Course 6.871 notes, MIT Spring 74.

[184] M Masson. Automatic program generation. In *Proceedings of the Eighth International Conference on Cybernetics*, International Association of Cybernetics, Namur, Belgium, Sep 1976. in French.

[185] H Matsunaga, T Mizuno, and T Ideguchi. Design and implementation of automatic program generation system based on the formal description of protocols. *Transactions of Information Processing Society of Japan*, 22(6):574–580, 1981. in Japanese.

[186] B P McCune. The PSI program model builder: synthesis of very high-level programs. In *Proceedings of ACM SIGART-SIGPLAN Symposium on Artificial Intelligence and Programming Languages*, pages 130–139, Aug 1977.

[187] W M McKeeman, J J Horning, and D B Wortman. *A Compiler Generator.* Prentice-Hall, Englewood Cliffs, NJ, 1970.

[188] G Mints and E Tyugu. Justification of the structural synthesis of programs. *Science of Computer Programming*, 2(3):215–240, 1982.

[189] L Molnar and P Navrat. System of automatic synthesis of programs with abstract data types. In *Tagungsbuch 30. Internationales Wiss. Kolloquium TH Ilmenau*, pages 155–158, Vortragsreihe Informatik, 1985.

[190] L Molnar and P Navrat. Using hints in automatic program synthesis. In *Proceedings of SOFASR 78*, pages 312–327, 1978.

[191] D J Mostow. *Mechanical Transformation of Task Heuristics into Operational Procedures.* PhD thesis, Carnegie-Mellon University, Pittsburgh, PA, 1981.

[192] P Navrat. *Automatic Program Synthesis and Methodology of Programming.* PhD thesis, Slovak Technical University, Bratislava, Czechoslovakia, 1983.

[193] J M Neighbors. *Software Construction Using Components.* PhD thesis, University of California, Irvine, CA, 1980.

[194] Office of Planning and Research. *Automatic Generation of Computer Programs for Converting Transmitter Data to IRS Standards—Phase I Report.* Contract TIR-77-62, Internal Revenue Service, Washington, DC, 1977.

[195] Office of Planning and Research. *MODEL II--Automatic Program Generator User's Manual.* Internal Revenue Service, Washington, DC, Jul 1978. Contract TIR-77-41.

[196] Mathematical Computing Advisory Panel. *Symposium on Automatic Programming for Digital Computers*, Office of Naval Research, Department of the Navy, May 1954.

[197] E Papon. *Algorithmes de Detection dc Relations de Recurrence - Application a la Synthese et a la Transformation de Programmes*. PhD thesis, Universite Orsay, 1981. These de 3eme cycle.

[198] L Paulson. *A Compiler Generator for Semantic Grammars*. PhD thesis, Stanford University, Stanford, CA, Dec 1981.

[199] J Penjam. A method of automatic implementation of compilers. *Cybernetics*, 4, 1980.

[200] J Penjam. Synthesis of semantic processors from attribute grammars. *System Programming Computer Software*, 1:50–60, 1983.

[201] A J Perlis. Automatic programming. *Quarterly Applied Math*, 30:85–90, 1972.

[202] F E Petry. *Program Inference from Example Computations Represented by Memory Snapshot Traces*. Technical Report OSU-CISRC-TR-75-1, Computer and Information Science Research Center, Ohio State University, Columbus, OH, 1975.

[203] F E Petry. *Program Inference from Example Computations Represented by Memory Snapshot Traces*. PhD thesis, Ohio State University, Columbus, OH, 1974.

[204] J V Phillips. Program inference from traces using multiple knowledge sources. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, page 812, Aug 1977.

[205] A Pnueli, N S Prywes, and R Zarhi. *Automatic Program Construction Techniques*, chapter 13. Macmillan, New York, 1984. titled Scheduling Equational Specifications and Non Procedural Programs.

[206] N Prywes. *Advances in Computers*, chapter . Volume 16, Academic Press, New York, 1977. titled Automatic Generation of Computer Programs.

[207] N Prywes and A Pnueli. Automatic program generation in distributed cooperative computation. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-14(2):275–286, 1984.

[208] N S Prywes, A Pnueli, and S Shastry. Use of a nonprocedural specification language and associated program generator in software development. *ACM Transcations on Programming Languages and Systems*, 1(2):196–217, 1979.

[209] J A Ramirez. *Automatic Generation of Data Conversion Programs Using a Data Description Language.* PhD thesis, University of Pennsylvania, Philadelphia, PA, 1973.

[210] P Raulefs. *Automatic Synthesis of Minimal Algorithms from Samples of the Behavior.* Technical Report INFI-73-2, Institut fur Informatik, Universitat Karlsruhe, Karlsruhe, West Germany, Jan 1973.

[211] T Reps and T Teitelbaum. The synthesizer generator. In *Proceedings of ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments*, pages 42–48, Apr 1984.

[212] C Rich. *A Library of Plans with Applications to Automated Analysis, Synthesis, and Verification of Programs.* Technical Report TR-604, MIT Artificial Intelligence Laboratory, Cambridge, MA, Jan 1982.

[213] C Rieger and P London. Subgoal protection and unravelling during plan synthesis. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 487–493, Aug 1977.

[214] N A Rin. *Automatic Generation of Business Data Processing Programs from a Non-Procedural Language.* PhD thesis, University of Pennsylvania, Philadelphia, PA, 1973.

[215] G-C Roman. An argument in favor of mechanized software production. *IEEE Transactions on Software Engineering*, SE-3(6):406–415, 1977.

[216] S Russell and A S Elmaghraby. An automatic program generator for dynamic simulation. In *Conference Proceedings of IEEE SOUTHEASTCON 84*, pages 111–114, IEEE, New York, Apr 1984.

[217] G Ruth. Automatic design of data processing systems. In *Proceedings of the Third ACM Symposium on Principles of Programming Languages*, Jan 1976.

[218] G Ruth. *Protosystem I: An Automatic Programming System Prototype.* Technical Report LCS-TM-72, Laboratory for Computer Science, MIT, Cambridge, MA, Jul 1976.

[219] K Sakaue and H Tamura. Automatic generation of image processing programs by knowledge-based verification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 189–192, IEEE Computer Society Press, Silver Spring, MD, Jun 1985.

[220] M Sato. Towards a mathematical theory of program synthesis. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 757–762, Aug 1979.

[221] W L Scherlis. *Expression Procedures and Program Derivation.* PhD thesis, Stanford University, Stanford, CA, 1980.

[222] A C Semeco, P Eamsherangkoon, S P Roth, and D Q Vu. The role of understnading in automated program synthesis. In *Proceedings of SPIE—Applications of Artificial Intelligence II*, SPIE—The International Society for Optical Engineering, Bellingham, WA, Apr 1985.

[223] S Shastry. *Verification and Correction of Non-Procedural Specifications in Automatic Generation of Programs.* PhD thesis, University of Pennsylvania, Philadelphia, PA, May 1978.

[224] D Shaw, W Swartout, and C Green. Inferring LISP programs from examples. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, pages 268–273, Sep 1975.

[225] L Siklossy. The synthesis of programs from their properties, and the insane heuristic. In *Proceedings of the Third Texas Conference on Computing Systems*, 1974.

[226] L Siklossy and D A Sykes. Automatic program synthesis from example problems. In *Advance Papers for the Fourth International Joint Conference on Artificial Intelligence*, pages 268–273, Sep 1975.

[227] C H Smith. Applications of a general theory of learning by example to automatic program synthesis. In *Proceedings of the International Conference on Systems, Man and Cybernetics*, pages 373–375, Dec-Jan 1983.

[228] C H Smith. The power of parallelism for automatic program synthesis. In *Proceedings of the 22nd Annual Symposium on Foundations of Computer Science*, pages 283–295, IEEE, 1981.

[229] D R Smith. *A Class of Synthesizable LISP Programs.* Master's thesis, Duke University, Durham, NC, Jun 1977.

[230] D R Smitl . *Sixth Conference on Automated Deduction*, pages 172–193. Volume 138 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1982. titled Derived Preconditions and their Use in Program Synthesis.

[231] D R Smith. A survey of synthesis of LISP programs from examples. In *Proceedings of the International Workshop on Program Construction*, Sep 1980.

[232] D R Smith. Top-down synthesis of divide-and-conquer algorithms. *Artificial Intelligence*, 27(1):43–96, 1985.

[233] D R Smith. *Top-Down Synthesis of Simple Divide-and-Conquer Algorithms.* Technical Report NPS-52-82-11, Department of Computer Science, Naval Postgraduate School, Monterey, CA, 1982.

[234] J M Spitzen. *Approaches to Automatic Programming.* Technical Report TR 17-74, Center for Research in Computing Technology, Harvard University, Cambridge, MA, 1974.

[235] R Sproull. *Using Program Transformations to Derive Line-Drawing Algorithms.* Technical Report CMU-CS-81-117, Carnegie-Mellon University, Pittsburgh, PA, 1981.

[236] C V Srinivasan. *Programming Over a Knowledge Base: The Basis for Automatic Programming.* Technical Report SPSAP-TM-4, Department of Computer Science, Rutgers University, New Brunswick, NJ, 1973.

[237] T A Standish. *Observations and Hypotheses about Program Synthesis Mechanisms.* Technical Report 278, Bolt, Beranek, and Newman, Cambridge, MA, 1973.

[238] E Subrahmanian and R L Cannon. A generator program for models of discrete-event systems. *Simulation*, 93–101, 1981.

[239] P A Subrahmanyam. An automatic/interactive software development system: formal basis and design. In A Wasserman, editor, *Proceedings of the IFIP WG8.1 Working Conference on Automated Tools for Information Systems Design and Development*, pages 125–146, North-Holland, 1982.

[240] P A Subrahmanyam. A basis for a theory of program synthesis. In *Proceedings of the First Annual National Conference on Artificial Intelligence*, pages 74–76, Aug 1980.

[241] P A Subrahmanyam. *A Formal Basis for Program Synthesis.* Technical Report, University of Utah, Feb 1980.

[242] P A Subrahmanyam. *Towards a Theory of Program Synthesis: Automating Implementations of Abstract Data Types.* PhD thesis, State University of New York at Stony Brook, Aug 1979.

[243] P D Summers. A methodology for LISP program construction from examples. *Journal of the Association for Computing Machinery*, 24:161–175, 1978.

[244] P D Summers. *Program Construction from Examples.* PhD thesis, Yale University, New Haven, CT, Dec 1975.

[245] W Swartout. XPLAIN: a system for creating and explaining expert consulting programs. *Artificial Intelligence*, 21:285–325, 1983.

[246] J-P Treuil, J-P Jouannaud, and G Guiho. *LQAS un Systeme-Question-Reponse Base sur l'Apprentissage et la Synthese de Programmes a Partir d'Ezamples.* Technical Report, Institut de Programmation, Universite de Paris VI. Mar 1977.

[247] E Tyugu. *Methods of Algorithmic Language Implementation*, pages 251–267. Volume 47 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1977. titled A Programming System with Automatic Program Synthesis.

[248] E Kh Tyugu. Computational frames and structural synthesis of programs. *Engineering Cybernetics*, (6):101–106, 1982.

[249] E Kh Tyugu. A system of programming with automatic construction of algorithms. In *Proceedings of the Symposium on Methods of Realization of New Algorithmic Languages*, 1975.

[250] J W Ulrich and R Moll. Program synthesis by analogy. In *Proceedings of Symposium on Artificial Intelligence and Programming Languages*, pages 22–28, Aug 1977.

[251] USC/Information Sciences Institute. *Automatic Programming.* Annual Technical Report SR-75-3, USC/Information Sciences Institute, Marina Del Rey, CA, Jun 1975. Page 24-?

[252] V Vojtek. *Contribution to the Automatic Program Synthesis.* PhD thesis, Slovak Technical University, Bratislava, Czechoslovakia, 1985.

[253] V Vojtek, L Molnar, and P Navrat. Automatic program synthesis using heuristics and interaction. *Computers and Artificial Intelligence*, 5(5):429–442, 1986.

[254] F W von Henke. On generating programs from data types: an approach to automatic programming. In *Proceedings Colloques IRIA sur la Construction, L'Amelioration et la Verification Systematique des Programmes*, 1975.

[255] R Waldinger. *Constructing Programs Automatically Using Theorem Proving.* PhD thesis, Carnegie-Mellon University, Pittsburgh, PA, 1969.

[256] R Waldinger and R Lee. PROW: a step toward automatic programming. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1969.

[257] D Warren. Generating conditional plans and programs. In *Proceedings of the AISB Summer Conference*, pages 344–354, Jul 1976.

[258] M Wegmann and M Baechler. A concept for the implementation of dialogue systems. *Output*, 11(4):31–35, 1982. in German.

[259] I Wladawsky. The mentor for business applications (MBA): a natural language automatic programming system. Mar 1975. Memo, IBM Thomas J Watson Research Center.

[260] R J Wood. *Computer Aided Program Synthesis*. Technical Report TR-861, University of Maryland, College Park, MD, 1980.

[261] W A Wulf. PQCC: a machine-relative compiler technology. In *Proceedings of COMPSAC 80*, pages 24-36, Oct 1980.

[262] R T Yeh, R Mittermeir, N Rossopoulos, and J Reed. A programming environment framework based on reusability. In *Proceedings of the International Conference on Data Engineering*, pages 277–280, IEEE Computer Society Press, Silver Spring, MD, Apr 1984.